Remarks/Arguments

With reference to the Office Action of April 24, 2006 Applicants offer the following remarks.

Non-Art Rejections

In the Office Action of April 24, 2006, Claims 28-50 were rejected under 35 USC §101, and claims 4, 5, 10-11, 14, 17, 22-24, 27-28, 31, 36-37, 44, 49, and 50 were rejected were 35 USC §112 (second paragraph). Appropriate amendment has been made.

Art Rejections

In the Office Action of April 24, 2006, claims 15, 7, 13, 15-32, 34-40, and 42-50 were rejected under 35 USC §102(b) as anticipated by U.S. Patent 6,425,075 to Stiles et al. for Branch Prediction Device With Two Levels of Branch Prediction Cache, and claims 6, 14, 33, 41 were rejected under 35 USC §103(a) as being unpatentable over Stiles, *above*.

The Art of Record

United States Patent 6,425,075 to Stiles et al. for Branch Prediction Device With Two Levels Of Branch Prediction Cache describes a branch prediction cache (BPC) scheme that utilizes a hybrid cache structure. The BPC provides two levels of branch information caching. The fully associative first level BPC is a shallow but wide structure (36 32byte entries), which caches full prediction information for a limited number of branch instructions. The second direct mapped level BPC is a deep but narrow structure (256 2byte entries), which caches only partial prediction information, but does so for a much larger number of branch instructions. As each branch instruction is fetched and decoded, its address is used to perform parallel lookups in the two branch prediction caches.

In brief, Stiles et al describe two levels of branch information caching. The first level BPC is a shallow (36 entries) but wide (entries are approximately 32 bytes each) structure which caches full prediction information for a limited number of branch instructions. The second level BPC is a deep (256 entries) but narrow (entries are approximately 2 bytes each) structure which caches only partial prediction information, but does so for a much larger number of branch instructions. As each branch instruction is fetched and decoded, its address is used to perform parallel lookups in the two branch prediction caches. (Siles et al., column 3, lines 27-38)

Applicants' Claimed Invention

Status of the Claims.

Claims 1-50 were originally presented for examination. All 50 claims were rejected; claims 15, 7, 13, 15-32, 34-40, and 42-50 were rejected under 35 USC §102(b) as anticipated by U.S. Patent 6,425,075 to Stiles et al., and claims 6, 14, 33, 41 were rejected under 35 USC §103(a) as being unpatentable over Stiles, *above*.

Applicants have extensively amended all of the independent claims (Claims 1, 24, and 28), and canceled twenty claims.

All of the remaining claims now contain the following claim limitations (eith directly or through dependency):

--- pipelining

--- pipelined processor has a branch target buffer (BTB)

--- creating a recent entry queue,

    --- the recent entry queue comprises a set of branch target buffer (BTB) entries in parallel with the branch target buffer (BTB),

--- organizing the recent entry queue as a FIFO queue,

--- comparing an entry to be written into the BTB against valid entries within the recent entry queue,

--- blocking an entry matching an entry within the recent entry queue from being written into the BTB,

--- searching the recent entry queue to detect looping branches,

--- comparing the branch to determine if it was recently written into the queue,

--- determining if the branch is backwards branching whereby a looping branch is detected, and if a looping branch is detected that is not predicted thereafter, delaying a decode, and

--- writing an entry into the BTB when it is also written into the recent entry queue.

Exemplary Claim

Claim 1, as amended, is exemplary.

> 1. (Currently Amended) A method of operating a computer having a pipelined processor having a branch target buffer (BTB) comprising creating a recent entry queue, said recent entry queue comprising a set of branch target buffer (BTB) entries in parallel with the branch target buffer (BTB), organizing the recent entry queue as a FIFO queue, comparing an entry to be written into the BTB against valid entries within the recent entry queue, blocking an entry matching an entry within the recent entry queue from being written into the BTB, searching the recent entry queue to detect looping branches, comparing the branch to determine if it was recently written into the queue, determining if the branch is backwards branching whereby a looping branch is detected, and if a looping branch is detected that is not predicted, and thereafter delaying a decode, and writing an entry into the BTB when it is also written into the recent entry queue.

Discussion: Art Rejection

The overarching issue is whether the claims, as a whole, as limited by the newly added clauses and limitations, are allowable over the art of record.

Claims 1-23 will be analyzed in detail.

Original claim 1 was rejected as anticipated by Stiles, Figure 2, where cache 155 is said to be a branch target buffer and cache 152 is said to be a recent entry queue in parallel with the branch target buffer. However, there is no specific disclosure that cache 155 is a branch target buffer and cache 152 is a recent entry queue in parallel with the branch target buffer.

Claim 1 has been amended by adding the limitations of claims 2 and 3. Claim 2 was rejected as anticipated by Stiles, column 3, lines 28-38[1] and Claim 3 was rejectd as anticipated by Stiles, column 10, lines 37-40.[2] However, Stile's disclosure of "two levels of branch information processing" and "performing parallel lookups in the first and second level BPCs" do not teach or suggest applicants' claimed "A method of operating a computer having a pipelined processor having a branch target buffer (BTB) comprising creating a recent entry queue, <u>said recent entry queue comprising a set of branch target buffer (BTB) entries</u> in parallel with the branch target buffer (BTB), <u>organizing the recent entry queue as a FIFO queue…"</u>

Applicants have next amended claim 1 adding the limitations of claims 5, 6, and 7, i.e.,

> --- comparing an entry to be written into the BTB against the valid entries within the recent entry queue.

---

[1] In brief, the invention provides <u>two levels of branch information caching</u>. The <u>first level</u> BPC is a shallow (36 entries) but wide (entries are approximately 32 bytes each) structure which caches full prediction information for a limited number of branch instructions. The <u>second level</u> BPC is a deep (256 entries) but narrow (entries are approximately 2 bytes each) structure which caches only partial prediction information, but does so for a much larger number of branch instructions. As each branch instruction is fetched and decoded, its address is used to perform parallel lookups in the two branch prediction caches. Column 3, lines 27-38

[2] In parallel with instruction decoding, the instruction's decode PC is used to <u>perform parallel lookups in the first and second level BPCs.</u> (Column 9. lines 35-37)

--- blocking an entry matching an entry within the recent entry queue from being written into the BTB.

--- when an entry is written into the BTB it is also written into the recent entry queue.

In the Office Action these were said to be anticipated by Stiles Column 10, line 41-67[3], buttressed by *"The examiner asserts the L1 BPC holds only recent branch data and the L2 BPC holds data covering a longer history of branches. Inherently, if branch data is to be retained longer then the L1 BPC can hold it, it must also be written into the L2 BPC."* and a discussion of arrangements of cache, but as conceded in the Office Action, Stiles fails to disclose blocking an entry matching an entry within the re cent entry queue from being written into the BTB.

Applicants have amended claim 1 by adding the limitations of claim 15-18, i.e.,

--- searching the recent entry queue to detect looping branches.

--- the branch to determine if it was recently written into the queue.

--- determining if the branch is backwards branching whereby a looping branch is detected.

--- first detecting a looping branch is detected that is not predicted, and thereafter delaying a decode.

---

[3] Given the relatively limited size of this first level cache, it is designed to support access in a highly associative manner versus a direct mapped or two/fourway set associative manner. This lookup, to check whether an entry currently exists in the cache for a branch about to be processed, is typically performed using the address of the branch instruction. For some pipeline designs, a closely related address may instead need to be used.

In this context, the term fully associative means that bits of the input address are compared with potentially matching bits of all the entries in cache. A hit is defined to have occurred when the input address matches at least one stored address. The term direct mapped means that some number of bits of the input address are used to index into the memory, and the remaining bits are stored in the memory. When the entry is read out of the memory, the stored address bits are compared with the corresponding bits in the input address, and a hit is defined to have occurred when the two agree. This entails a single comparison. In the event that two branch instructions index to the same entry (different PC's with the same set of index bits), a direct mapped cache can only accommodate one, even if there are other empty entries. The term twoway set associative refers to a structure with two memory banks, so that two entries are read out for a given index and two comparisons are performed. This concept can be expanded to more than twoway set associative. (Column 10, lines 41-67)

It is said in the Office Action that these limitations are anticipated by assertions that "*The examiner asserts that the L1 BPC is searching to detect any branch entry including looping branches.*" The next assertion is that "*The Examiner asserts that if any entry appears in the L1 BPC, it was recently written*". And, "*The Examiner asserts that if the stored target address is of a lower value than the current PC address, the branch is inherently identified as a backwards branch.*" There is also one citation to Stiles et al., column 16, lines 15-18[4]

Stiles, with and without the added assertions of the Examiner neither teaches nor suggests the invention of Applicants' amended claim 1, certainly not with the specificity required of a 35 USC §102(b) anticipation rejection or a 35 USC §103(a) obviousness rejection.

Discussion: 35 USC §101 (Not Statutory Subject Matter)
and 35 USC §112 (Second Paragraph) Objections and Rejections

Applicants have amended the claims to overcome the 35 USC §101 and 35 USC §112 (Second Paragraph) rejections and objections.

[4] Of course, when one or more of the predictions is subsequently found to be wrong, any hidden delays become exposed and possibly an additional delay incurred while the CPU pipeline is flushed and/or restored to an appropriate state from which to continue correct instruction processing. **(Column 16, lines 15-18)**

Conclusion

Based on the above discussion, it is respectfully submitted that the pending claims describe an invention that is statutory subject matter and is properly allowable to the Applicants.

If any issues remain unresolved despite the present amendment, the Examiner is requested to telephone Applicants' Attorney at the telephone number shown below to arrange for a telephonic interview before issuing another Office Action.

Applicants would like to take this opportunity to thank the Examiner for a thorough and competent examination and for courtesies extended to Applicants' Attorney.

Respectfully Submitted

Certificate of Mailing

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as Certified Priority Mail (Return Receipt Requested), **Certified Label 7006 0810 0001 1252 2546**, in an envelope addressed to the Commissioner for Patents, Alexandria Virginia, 22313

Date of deposit: October 21, 2006

Person mailing paper: Richard M. Goldman

Signature:

Richard M. Goldman, Reg. # 25,585
371 Elan Village Lane, Suite 208
San Jose, CA 95134
Voice: 408-324-0716
Fax: 408-324-0672
E-mail: goldmanptn@aol.com